Multiple-Domain Correlation Without
Highly Structured Standardization of Data

Aaron Hutchinson
Cogent3D, Inc.
Tucson, AZ
Aaron@Cogent3D.com

Nephi Lewis
Rockwell Collins
Salt Lake City, UT
Enlewis@RockwellCollins.com

ABSTRACT

Many in the simulation industry believe the only way to assure correlation across multiple simulation training domains (e.g., space to urban) is to structure highly the source data used to build those training scenes. This paper will talk about the use of new real-time concepts to attain correlation across multiple simulation domains by directly using source data in its native format found scattered on your disk or network, instead of the elaborate, and often restrictive "common/master" formats being proposed worldwide.

The paper will point out in parallel new concepts of implementing and conceptualizing the multiple domain correlation problems by employing real-time construction techniques to obtain the next level of multiple application reuse, scalability, and correlation.

Important to the paper will be the discussion that source data alone, no matter how many attributes or how elaborately stored, is not enough to assure a multiple-domain, or even a multiple-system, correlated representation, with efficient use of resources, and proper representation in various systems. Because many of the decisions (that really are better determined in real-time) are statically compiled into the 3D terrains and cannot be accounted for by selecting a different level-of-detail (LOD) from a large selection of pre-built static LODs. An example of an important dynamic variable is screen resolution, which is used in the calculations to determine the number of LODs and the ranges at which 3D object, terrain, and imagery transition in/out of the 3D scene, so they do not create negative training by distracting the student by popping in/out.
Discover how new concepts fueled by multiple CPUs/GPUs come together to create 3D scenes completely on the fly for multiple domains and how building terrain completely on the fly opens up the doors for new opportunities in cost savings, capabilities, and scene fidelity across multiple domains at the same time.

ABOUT THE AUTHORS

Aaron Hutchinson is President of a new small business (Cogent3D, Inc.) that specializes in products and services to solve complex 3D problems. Previously, he was an executive at Diamond Visionics, LLC, Presagis USA, and Terrain Experts, Inc (TERREX). Before that he was Senior Visual Engineering at NAVAIR Orlando working for Jimmy Burns and alongside Pam Woodard. He is a graduate of University of Central Florida (UCF) with a B.S.E. in Computer Engineering.

Nephi Lewis has been with Rockwell Collins STS (and formerly, Evans & Sutherland Computer Corp.) for over 25 years, and is currently a principal product manager for EP Real-Time, Modeling Tools, and Synthetic Environment products. Previously, Mr. Lewis served as manager of Software Testing and QA, 3D Content Development for E&S's Digital Studio group, and EPX real-time software development. He has experience in database design, technology exploitation, and software development for a wide variety of simulation applications: fast jet, attack jet, helicopter, mission rehearsal, ground warfare, and database products that serve as scenery for television and video. He is experienced with a number of platforms: CT-5A, CT-6, ESIG-x products, and EP-based products.

Mr. Lewis has published papers for I/ITSEC, IMAGE, and other conferences, as well as authoring a number of articles on various simulation and training topics. From 1994-97, Mr. Lewis taught the professional development course, "Database Modeling for Realtime Simulation" at the IMAGE Society's bi-annual conference and annual symposia.

Mr. Lewis graduated magna cum laude in Design Engineering from Brigham Young University in 1983, and has also completed graduate courses in advanced computer graphics from that institution, and graduate courses in business administration from the University of Utah.

Multiple-Domain Correlation Without Highly Structured Standardization of Data

Aaron Hutchinson
Cogent3D, Inc.
Tucson, AZ
Aaron@Cogent3D.com

Nephi Lewis
Rockwell Collins
Salt Lake City, UT
Enlewis@RockwellCollins.com

Background

Structured Data for Reuse, Not Runtime

SIF, SEDRIS, and NPSI/CDS are just a few Government-sponsored formats designed to improve the reuse of enhanced and corrected source data in a variety of training and simulation applications. By standardizing on a small number of industry-common formats and attribute combinations, much data processing, reconciliation, and quality-control work is shared, thereby lowering database production costs. But the main focus is not improving correlation, (though there are some by-products in that area), but the elimination of waste in preparing, sharing, and evaluating data. All vendors benefit by converging on a set of commonly-used and -understood data formats. Proprietary formats are obviated; they add no real value to the process at this point, anyway.

Structured Data for Runtime

OTF, CTDB, LADBM, TerraPage, IVE, MetaFlight, and CDB are just a few formats designed to improve *real-time* paging, culling, and rendering/searching performance. The object of these formats is the offload of as many tasks as possible from the real-time simulation system by doing them offline and in advance, while storing the results in a structured format, so that lower-performing (and lower-costing) hardware and software can be used to perform the rendering/searching tasks. This was an important process when purchasing CPU and rendering performance was expensive. CPUs of today, however, are no longer the bottleneck in most training and simulation applications. The latest versions are now out-performing the multiple-million-dollar Cray, SGI, and Mainframe systems of not that long ago.

Multiple Domain Simulation

What is Multiple Domain Simulation? Actually it is not one thing; Multiple Domains can be any of the following:

a) Sensor, visual, and SAF simulations inside a single simulation device.
b) Space, air, maritime, ground, and urban simulations across multiple devices tied together in a single simulation.
c) Live, virtual, and constructive (LVC) training on the same, or different, platforms interacting with one another in the same scenario.
d) Simulation, training, and operational; the correlation of systems across applications or intent.
e) Or any combination of a, b, c, and d.

What is Correlation?

The first visual systems had few cross-domain, or multiple-system correlation worries. However, today's SAF, RADARs, sensors, and visual systems need to be flexible enough to present correlated scenes for any number of domains (Space, Air, Maritime, Urban, and Ground) as well as LVC and operational applications. So what is this correlation that everyone talks about?

- Correlation *inside* a system means the same scene is represented consistently in all channels; viewers see the same features, same terrain, and if appropriate, the same imagery (within the context of that portion of the EM spectrum being rendered). In this space, relative correlation (or content representation) among scenes within the simulator is more important than correlation with the real world. Thus, consistent representation is more important than accuracy.
- Correlation *across* systems means all the same features, terrain, and if appropriate, imagery from their relative perspective. A road, a building, or a tree might be represented slightly differently from a space simulation perspective versus a

helicopter's perspective versus a tank simulation perspective, but will all be at a consistent location and represented with consistent, range-appropriate details.

- Correlation in LVC is another matter altogether; the simulations need to represent the real-world as exactly as possible with both positional and content accuracy (absolute accuracy). Unlike simulation-only environments, one cannot reduce or change the real-world to correlate with simulation-rendering limitations.
- Correlation in simulation, training, and operational means more than terrain content: it means tactical feeds, hardware/software interfaces, timeliness of data, and much more information about the terrain and weather, but also about the entities running on, above, and below the terrain.

But correlation is also training-context dependent. It is not always an absolute; what does it mean for something in the scene to "be in the same location?" Clearly, the answer will be different for a KC-135 boom operator trainer versus a man-wearable device used by a platoon leader. Not only must data be accompanied with tolerance information, but some measure about how realistically the object is represented in the scene. As an example, here are two images of the same "thing" in the scene; a road on side of a hill. One road has been cut and filled into the terrain surface, while the other has not been. There is nothing wrong with either representation, when used in an appropriate context, but merely storing data about the scene in the same format does not guarantee that any particular definition of correlation will be met by the dataset. If dataset standards are too restrictive, or tied too closely to any one application, it will be difficult, or impossible, for other vendors to add technology-pushing features to new training devices.

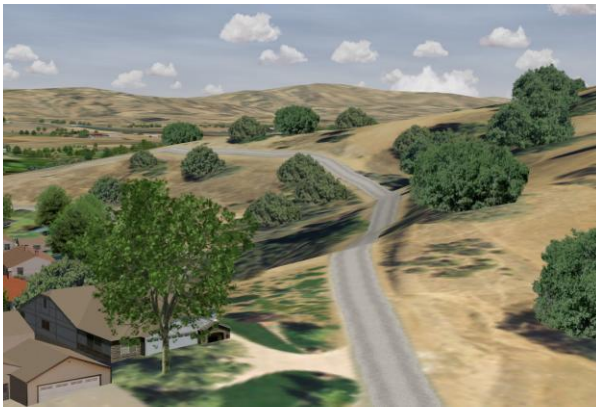

Fig 1: Gravel road – no cut and fill

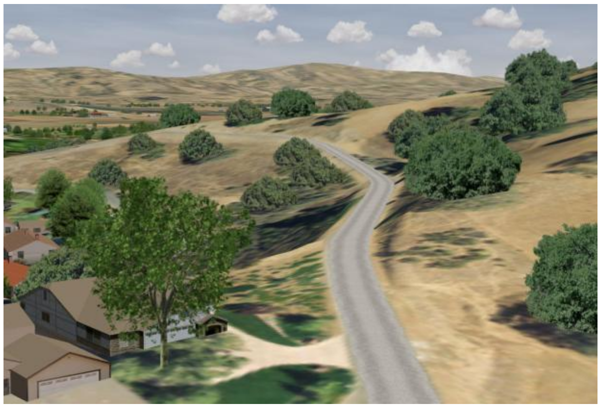

Fig 2: Gravel road – with cut and fill

If one vendor's system can automatically create cut and fill for a feature, should the data standard dictate how that feature is to be added to the scene? To do so might limit scene realism by forcing some features to go unused, because the implementation doesn't comply with the employed data "standard."

Highly Structured Data

Isn't highly-structured data required for acceptable correlation?

Many today are advancing the notion that one must use highly-structured data to achieve a proper level of correlation. Falling into this type of thinking is easier than one might think. Correlation among various sub-systems of a trainer has suffered in the past due to several reasons: a) There is seldom enough time to build all the formats completely to the same revision in a given program's schedule; b) The format or system in question did not support the entire required dataset; or, c) Not enough time was available to transfer and/or load the entire data package in the multiple required formats for several simulators being used in an event across large geographical distances.

The first logical conclusion might be that if all sub-systems just used the same format, never again would there be a correlation issue. The second logical conclusion might be that any single format able to support this would need to be a large, "includes-everything" solution. Once one makes this leap, it is a natural step to believe that to achieve correlation across multiple systems or within a system, one needs to control as much of the system as possible, by standardizing the formats, attributes, algorithms, supported data types, and anything else that might impede correlation. Government agencies have a phrase for this type of system: Stove Piped. Others call it closed or proprietary.

This approach is being marketed today in several forms; most are highly-structured formats that are targeted to one or another end application. While they attempt openness, they almost cannot help but cut off third-party innovations (and competition), as they are unable to anticipate future data requirements, or even data requirements of current applications, either because they are products of competitors, or include applications outside their ken. The format then becomes a barrier to entry for other players that might have otherwise introduced new technology and more effective training capabilities.

Consider the use of shadows in real-time rendered scenes. What if the industry (or even one segment of the industry) had adopted a format that excluded the necessary data for rendering real-time shadows a decade ago? What new features in our trainers might be delayed in the future by adopting too strict of a data format today?

Does highly structured data harm me?

Yes. Highly structured data does harm you. Here are several examples:

- The number of tools available to produce and use the structured format is limited (no matter how “open” the format is advertised to be), so competition is lower; costs will be higher.
- Highly-structured formats, by their nature, are optimized for particular sets of assumptions about data production and end-use applications. Trainer technology is rapidly changing and many assumptions quickly become obsolete. Current assumptions embedded in some formats today include filtering data into specific levels of detail (LOD) definitions, tying terrain, imagery, and vector LODs together, predefining paging sizes, and pre-MIPping information due to limited computational and I/O in past systems. (While at the same time Intel, AMD, NVIDIA, and others are completely redefining computational and I/O standards to overcome these legacy rendering problems.) Users of such dataset formats will soon need to convert their existing archives into new representations of the data, because the format standard is too restrictive, too detailed, and doesn’t address the heart of what is needed to define correlation. This results in a growing, overly-complex archive that just ends up duplicating costs.
- Highly structured data formats can optimize use, but at the expense of more complex, costly, and time-consuming preparation tasks. Important time is lost in this approach for time-critical targeting and 3D-enhanced tasks.
- Storing more than the original source data is a waste of resources. It takes longer to write data into the highly-structured format and longer to copy the changes to multiple sites, compared to more abstract forms of the data, that can be mathematically expanded later in the process.
- Periodic updated source materials are harder to integrate into a highly-structured data format because one must first find the original (now obsolete) information so it can be removed (DAFIF, DVOD, DNC, etc.).
- Standardizing on highly-structured 3D-simulation formats could mean needing to build a parallel structure to the one you use today for storing, moving, and managing existing GIS information both on in-house and deployed systems.
- Using highly-structured simulation-centric formats may mean having to augment all operational systems to support these new formats and interfaces to continue to have correlation across all your uses.

Overly-structured, or too-rigidly-defined formats can cost the user in lost time, duplication of efforts, and parallel capital systems. And still correlation may not be guaranteed, or even improved.

Can correlation happen any other way?

Yes, correlation is possible without the need to create a single large, “be-everything-to-everyone” format with thousands of restrictions applied or implied.

Correlation of multiple systems is not a simulation-unique problem. Our military services deal with correlation every day. Consider the coordination of air support with ground forces during an attack. Intelligence first *find* the target, ground forces next *confirm* the target, mission planning then lays out the attack plan, and aviation assets apply the weapon on the indentified threat. This process does not involve just one system accessing some single master format, but uses common assumptions about the earth model (like projection), and about the source data to achieve their goal. Adopting these assumptions, simulation systems and LVC events become easier, and the user gets to take advantage of more of the existing infrastructure already in place and in use.

Today, with the increase in performance of PC CPUs, I/O busses, and GPUs, vendors have incredibly more power available to perform tasks that formerly were impossible in real-time. Real-time systems today can emulate many of the non-real-time production tasks that are required to feed many of the highly-structured data formats assumptions. Examples:

- Creating alternative resolution representations of the source data. E.g., sampling five-meter imagery from one-meter original source imagery; moving vector information from one projection (UTM) to another (WGS84); feathering multiple-resolution images together in the same scene.
- Filtering information based on attributes or mathematical algorithms. For example, removing all data older than five months; selecting all 'pine tree' attributed features, or all features with a header label of 'No-Foreign.'
- Expanding the representation of information using source data, procedural, and other mathematical algorithms. Examples would include: increasing the number of trees scattered inside a defined area from 500 to 5000 per sq km; parametrically building a bridge; lofting 2D GIS data into 3D buildings.
- Correcting the source information based on availability of multiple source data files. E.g., smoothing holes and spikes in a high-resolution elevation files with older, or lower-resolution elevation files; ignoring vector features in erroneous locations.

As you can see, the more natural language to apply is math and Boolean attribution filtering. Instead of TINs (triangle irregular networks), use a gridded representation which is friendlier to the current and future rendering, physics, and path finding platforms/application. Grids make it much simpler to explain which posts to display in terms of mathematics, if they all are not used in the terrain calculation. The benefit of this approach is that the same copy of the information is used, culled, expanded, and represented; however, vendors are free to exploit any technical innovations to perform these tasks. Suppliers such as Cogent3D/Diamond Visionics, Rockwell Collins, Lockheed Martin, CATI, or any other vendor can have dynamic terrain systems which exploit various parts of the PC performance capability with their own optimizations, but still arrive at a level correlation in the simulated representation.

Can being unstructured harm me?

Yes, having structure *is* a balancing act. Not having enough structure is like the Wild West; this can adversely affect cost and correlation. But having too much is restrictive to creativity, increasing realism, and adds costs that might have otherwise been mitigated by newer technologies.

- Agreeing on common formats for storing information is positive. Limiting the number of formats allows for focused optimization of loading and processing the data. It also limits the number of cross-format conversions which take time, but add little or no value.
- Agreeing on common source data projections is good. This allows vendors to standardize many aspects of production; less work has to be done to use information.
- Agreeing on a common attribution is always good no matter if the source data will be slightly structured, highly structured, or not structured at all. Speaking and understanding the same language means that over time everyone's understanding of this language improves and tools can be pre-configured to operate on the source data with less end-user involvement.
- However, the problems start with storage and access. Agreeing on a common storage container or format is good, but it should be flexible and not tied to any one vendor, market, or business model. The formats listed in this paper are specific to the Modeling and simulation market. This is a problem because simulation systems do not drive operational systems like tanks, airplanes, and C2 (command and control), but instead play a supporting role.

"Train as You Fight" means more than it should look the same or act the same, but also means using the same information available to the troops. Source data used by the operational systems should be the starting point for simulation and LVC environments. For the US Army this means US Army Enterprise GIS system which is the ESRI GeoDatabase. Apply the missing simulation formats (OpenFlight) and attributions and you have a common operational and simulation environment that could be used by facilities, logics, planning, command and control, targeting, and training. By recognizing all the data use requirements early in the process, data movement and processing can minimized throughout the processes, thereby reducing cost and the complexity of learning and deploying the system.

Why take the risk? Dynamic Construction makes data preparation so much simpler and more flexible

Dynamic construction of training and simulation elements is a true paradigm shift in technology and how one thinks about the problem space. It is not just an evolution of the current methods.

Gone is the need for complex, highly-structured simulation specific formats that date back to the early days of simulation, just to assure that every polygon that *could* be used, *was* used. This means less complexity, fewer tools, less storage, less configuration management, and less cost.

Gone is the need for complex and expensive purpose-built tools that convert generic GIS data into specific forms of simulation-specific data to be stored in highly-structured simulation specific formats.

These are replaced by applications performing the same level of processing, but instead of applying it to all the data at once (just in case you want to fly over it one day), applying it only to the data that is actively being used (rendered) at the moment. And doing this in less than a second (16.66ms or 60Hz). This also means no unnecessary processing is being performed to the source data.

In a dynamic construction paradigm, the processing is done at the leaf/edge of the organization so it naturally expands as more users and uses of the technology emerge. Instead of being centralized, causing power plays and growth issues to support a growing user base. This means the same source data and application can produce a number of different, yet correlated representations of the source that are displayed uniquely to the users' requirements. (i.e., 3D models without textures, colored by attributes, ground areas colored by type, etc.)

The representation and resolution of one type of the source data is not tied to the representation and resolution of other types of source data. Thus you can have high-resolution terrain with low-resolution features or high resolution feature density with low resolution imagery or only information that meets some type of requirement, such as only NATO-provided information.

Dynamic Construction means built-in dynamic editing

We all know the old saying, "I will know it when I see it." This applies to simulation customers as well. Elaborate systems have been put in place in a great number of simulation integration companies to capture, track, and correct simulation DRs (deficiency reports). DRs are the industry's way of tracking problems they find in their trainers so that the customer knows what items need to be fixed, a contractor can fix those items, and so the customer can accept those changes sometime in the future.

Dynamic terrain brings advantages over existing systems with highly-structured static formats in that the customer can interactively modify the run-time database to fix problems, add missing data, and achieve the desired scene appearance. And then storing those changes back into the GIS system to be propagated, tracked, and configuration managed by the enterprise level GIS system.

Dynamic scene construction increases the ability to defer commitment

One of the seven tenets of Lean software development, a proven method for lower software development and maintenance costs, is that of "deferring commitment." At first this may sound counterintuitive, but it means that commitments in implementation should be deferred as long as possible so that greater flexibility in the final product can be maintained much later in the process. This permits developers and implementers to modify the trainer in response to "late-breaking" requirements. This also recognizes the fact that some requirements are not even clearly understood until portions of the system are implemented.

Dynamic scene construction enables end-users to insert requirements later in the process without adding nearly as much cost. Dynamic construction greatest strength is the ability for the user to be less aware of how the application works and more on the content.

Non-dynamic systems, and the formats that support them, have rules that must be followed to the letter or they will not perform optimally. They require you to understand the dynamics of image generation or some other 3D domain. Pageable blocks, levels of detail, range rings, clipping planes, and transitions are all vocabulary that needs not be understood by users of dynamic construction rendering systems. Dynamic construction systems abstract users from these worries. Using powerful real-time software and PC CPU resources to determine the best possible combination of variables frees the end user from a number of trade off decisions – or at least puts him or her in control of those decisions without requiring highly technical expertise.

Using a combination of screen resolution, source data resolution, and real-time performance monitoring, dynamic construction systems are able to find optimum configurations automatically, and provide the average user results that were once only possible by experts with many years of experience, an eye for performance, and a hefty budget that supported long integration timelines.

Dynamic construction systems automatically, in real-time, set the transition range of every 3D object in the scene, based on its size and the current screen rendering resolution. Static systems, which would embed the transition ranges, require hours of reprocessing and re-validation for each targeted screen resolution.

Dynamic construction systems store only the highest resolution of the elevation and imagery information and automatically create the lower representation of the elevation and imagery data based on screen resolution as needed to conserve disk/memory bandwidth and disk storage. Static systems sample all the elevation and imagery source data into all possible resolutions in case they are used.

Dynamic construction systems modify the elevation data based on vector information in real-time as needed, flattening lakes and rivers, cutting and filling around roads, smoothing out building foundations, and other elevation-based modifications. These modifications are applied to the highest resolution and newest elevation data available. Static systems pre-back the vector modifications into the elevation files, store them, and require redundant re-backing with the introduction of modified or updated data.

Fewer steps, less cost, and more flexibility while being easier, faster, and offering better results for everyone is the right direction, even if the first generation of this new technology does not do everything the aging products of the previous generation do today. With time, those features will be implemented, but more importantly, because data will be dynamic, the future holds even more capabilities that have not been possible in any version of the current, mainstream applications and we have yet to discover which will revolutionize training and operations.

REFERENCES

CAE USA, Inc. (December 2008). Common Database (CDB) Version 3.0 for USSOCOM. CDB 3.0 – Eight Draft.

CAE USA, Inc. (December 2006). USSOCOM Common Database (CDB) SE Core Master Database (MDB) Similarities, Differences & Synergies. IITSEC 2006 Presentation.

Riner, Bruce (December 2007). Quality Assurance and Standards for NPSI Datasets. IITSEC 2007 Paper.

Poppendieck, Tom and Mary, Implementing Lean Software Development: From Concept to Cash, pp 186, September, 2006, Addison-Wesley Professional, 1st edition.